

Martin Schwaighofer, MSc
martin.schwaighofer@ins.jku.at
Johannes Kepler University Linz, Austria

Case Study: Building and testing programming assignments with Nix

We use Nix to build and test student assignments for a systems programming course that we teach.

Many tools have been used by many teachers to automate this kind of task. Some tools support one or more specific programming environments. We speculate that many teachers simply write their own automation scripts, specific to the programming environment they teach in their course.

Using Nix we can decouple the supported programming environments from the automation around them, so that a user can freely specify a programming environment and a tool can provide some artifact, automated workflow or service that uses or provides the specified environment.

We have used Nix to build several teaching tools around the same declared environment:

- A grading pipeline which ingests all submissions for a specific assignment after they have been submitted, runs the tests for them and then outputs build logs and test results. This output is then handed over to several tutors for manual inspection.
- A self-updating VM tied to a repository on GitHub, which students can download and use to work on their assignments.
- A working prototype of a Go-based web service where students can upload their assignment submission and get immediate feedback while they are still actively working on it.

We have already employed the first two of those processes in practice during the last term.

What is interesting about building tools like this is that building them is about executing build steps in a controlled manner (e.g. when to perform specific steps, how to treat the result) as much as it is about building a precisely defined result.

For individual build steps we want properties like the following:

- We want to tolerate when they fail, proceed with the build and see the specific failure reflected in the output.
- We want to have a timeout be one configurable failure mode of a build step.
- We want to process student assignments without storing them.
- We want to treat build logs and test results as build outputs.
- We want to make builds inside and outside the build sandbox behave consistently.
- We want visibility of which parts of the build we break and fix while we work, in order to make it easier to improve tests iteratively.
- We want to prevent specific build outputs from being in the runtime closure of other build outputs to make sure we do not leak information through the live feedback web service.

During our talk:

- We would like to briefly introduce what our tools do for context and motivation.
- We would then pick a few interesting properties from the above list and present our various attempts to achieve them as users of Nix
- In between we would gather different approaches to individual problems we might have not thought of from the audience.

We do have solutions to get some of those properties, but some of our solutions are terrible hacks. We are also still actively working on this project, so we expect to make improvements to some of our current solutions until then.

We think that discussing needs that arise when writing tools like this, where individual build steps require additional control and the user cares about other outputs of the process than the build result, would be beneficial for the Nix ecosystem.

This work has been carried out within the scope of Digidow, the Christian Doppler Laboratory for Private Digital Authentication in the Physical World, funded by the Christian Doppler Forschungsgesellschaft, 3 Banken IT GmbH, Kepler Universitätsklinikum GmbH, NXP Semiconductors Austria GmbH, and Österreichische Staatsdruckerei GmbH.

(598 Words)

Video of Live Feedback Web Service POC:

<https://cloud.ins.jku.at/index.php/s/mq5XDpLfCEW3faj>

VM Repository:

<https://github.com/mschwaig/sysprog-vm>

Grading Pipeline and Live Feedback source (Repository still private):

<https://github.com/mschwaig/sysprog21-automation>

I'd be happy to give access to the private Repository on GitHub or via other means, please contact me. It's not fit for public consumption yet.